COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

University of Massachusetts Amherst. Spring 2020.

Lecture 18

LOGISTICS

- Problem Set 3 is due Monday 4/13 at 8pm.

SUMMARY

Last Class: Applications of Low-Rank Approximation

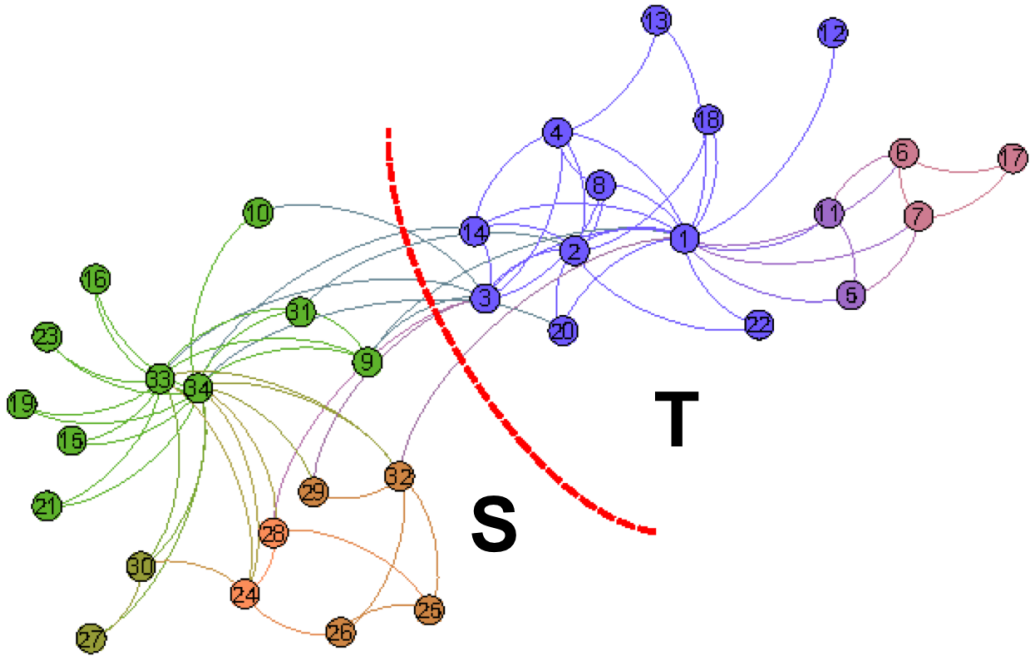
- Low-rank matrix completion (predicting missing measurements using low-rank structure).
- Entity embeddings (e.g., LSA, word embeddings). View as low-rank approximation of a similarity matrix.
- Start on spectral graph theory.

This Class: More Spectral Graph Theory & Spectral Clustering.

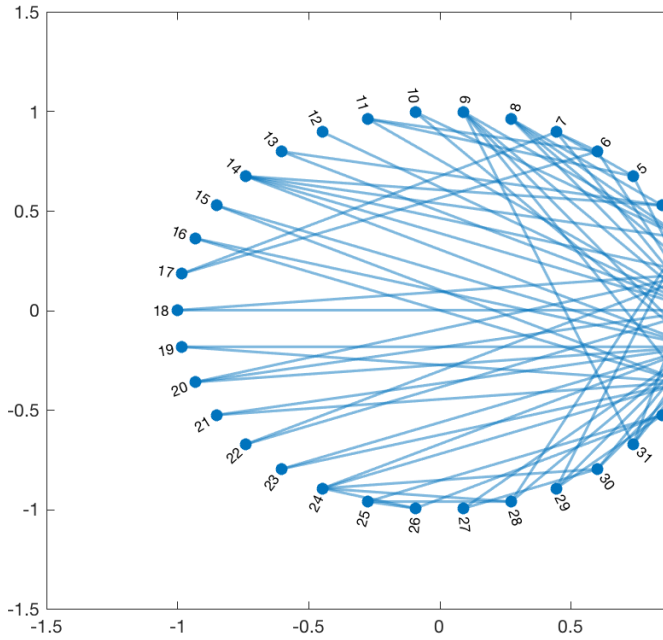
- Using eigendecomposition to partition graphs into clusters.
- Clustering non-linearly separable data.
- Application to the *stochastic block model* and community detection.

A very common task is to partition or cluster vertices in a graph based on similarity/connectivity.

Community detection in naturally occurring networks.



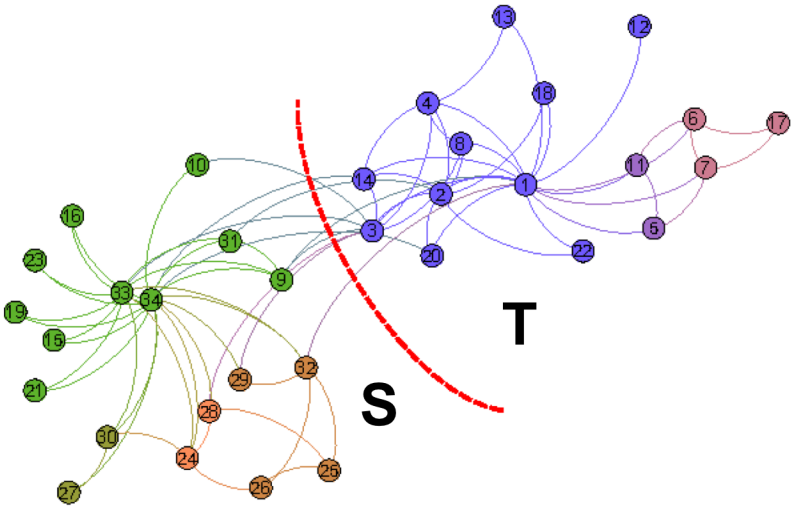
(a) Zachary Karate Club Graph



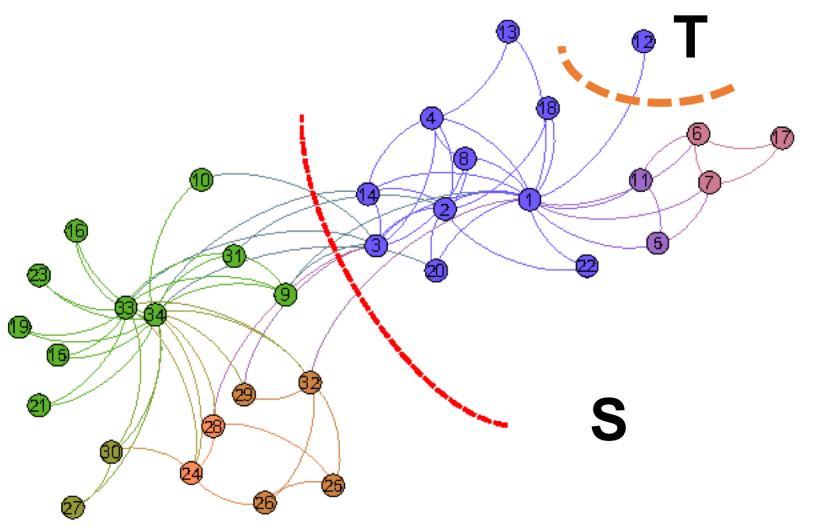
Linearly separable data

CUT MINIMIZATION

Simple Idea: Partition clusters along minimum cut in graph.



(a) Zachary Karate Club Graph



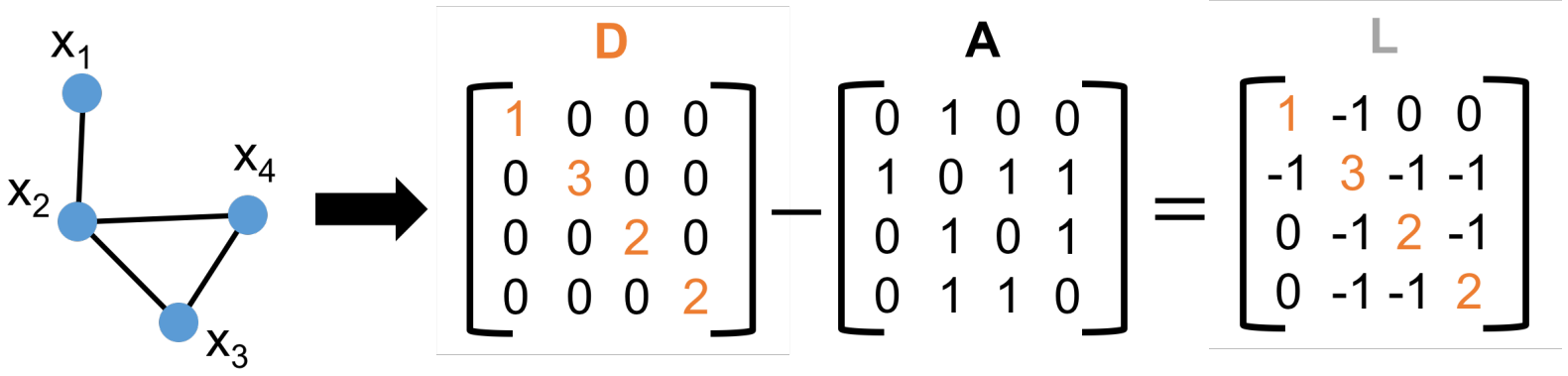
(a) Zachary Karate Club Graph

Small cuts are often not informative.

Solution: Encourage cuts that separate large sections of the graph.

- Let $\vec{v} \in \mathbb{R}^n$ be a cut indicator: $\vec{v}(i) = 1$ if $i \in S$. $\vec{v}(i) = -1$ if $i \in T$. Want $\vec{v}$ to have roughly equal numbers of 1s and -1s. I.e., $\vec{v}^T\vec{1} \approx 0$.

For a graph with adjacency matrix **A** and degree matrix **D**, $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the graph Laplacian.



For any vector $\vec{v}$,

$$\vec{v}^T L \vec{v} = \vec{v}^T D \vec{v} - \vec{v}^T A \vec{v} = \sum_{i=1}^{n} d(i)\vec{v}(i)^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} A(i,j) \cdot v(i) \cdot v(j)$$

For a cut indicator vector $\vec{v} \in \{-1, 1\}^n$ with $\vec{v}(i) = -1$ for $i \in S$ and $\vec{v}(i) = 1$ for $i \in T$:

1. $\vec{v}^T L \vec{V} = \sum_{(i,j)\in E} (\vec{v}(i) - \vec{v}(j))^2 = 4 \cdot cut(S, T)$.
2. $\vec{v}^T \vec{1} = |V| - |S|$.

Want to minimize both $\vec{v}^T L \vec{v}$ (cut size) and $\vec{v}^T \vec{1}$ (imbalance).

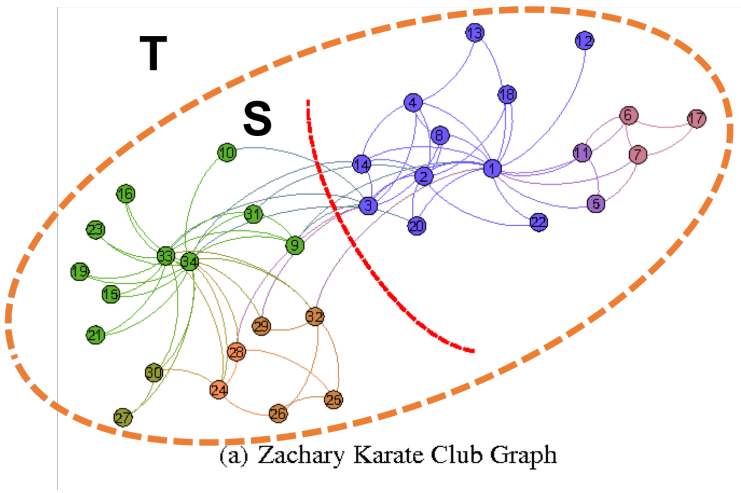
Next Step: See how this dual minimization problem is naturally solved by eigendecomposition.

SMALLEST LAPLACIAN EIGENVECTOR

The smallest eigenvector of the Laplacian is:

$$\vec{v}_n = \frac{1}{\sqrt{n}} \cdot \vec{1} = \underset{v \in \mathbb{R}^n \text{ with } \|\vec{v}\|=1}{\arg\min} \vec{v}^T L \vec{V}$$

with eigenvalue $\vec{v}_n^T L \vec{v}_n = 0$. Why?



(a) Zachary Karate Club Graph

n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$.

SECOND SMALLEST LAPLACIAN EIGENVECTOR

By Courant-Fischer, the second smallest eigenvector is given by:

$$\vec{v}_{n-1} = \underset{v \in \mathbb{R}^n \text{ with } ||\vec{v}||=1,\ \vec{v}_n^T \vec{v}=0}{\arg\min} \vec{v}^T L \vec{V}$$

If $\vec{v}_{n-1}$ were in $\{-1, 1\}^n$ it would have:

- $\vec{v}_{n-1}^T L \vec{v}_{n-1} = cut(S, T)$ as small as possible given that $\vec{v}_{n-1}^T \vec{1} = |T| - |S| = 0$.
- I.e., $\vec{v}_{n-1}$ would indicate the smallest perfectly balanced cut.
- The eigenvector $\vec{v}_{n-1} \in \mathbb{R}^n$ is not generally binary, but still satisfies a 'relaxed' version of this property.

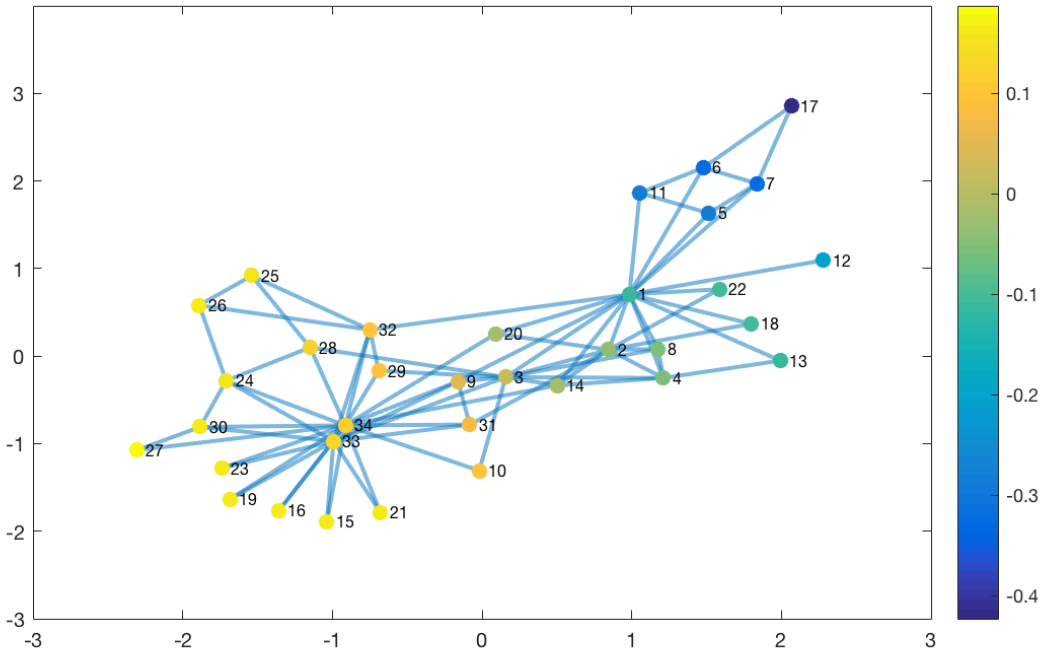
n: number of nodes in graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$: adjacency matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$: diagonal degree matrix, $\mathbf{L} \in \mathbb{R}^{n \times n}$: Laplacian matrix $\mathbf{L} = \mathbf{A} - \mathbf{D}$. S, T: vertex sets on different sides of cut.

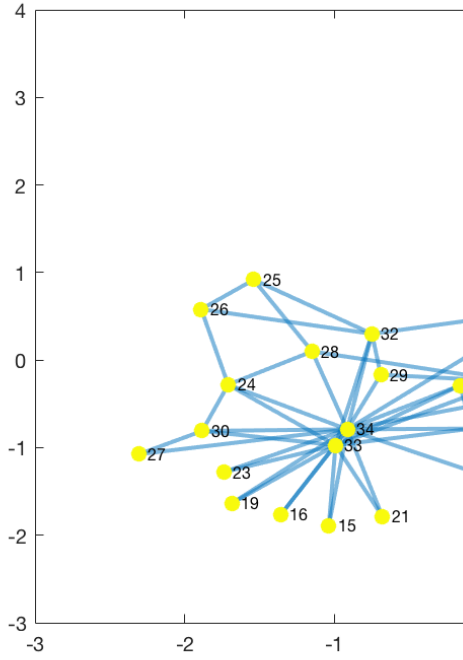
CUTTING WITH THE SECOND LAPLACIAN EIGENVECTOR

Find a good partition of the graph by computing

$$\vec{v}_2 = \underset{v \in \mathbb{R}^d \text{with } \|\vec{v}\|=1,\ \vec{v}_2^T\vec{1}=0}{\arg\min} \vec{v}^T L \vec{V}$$

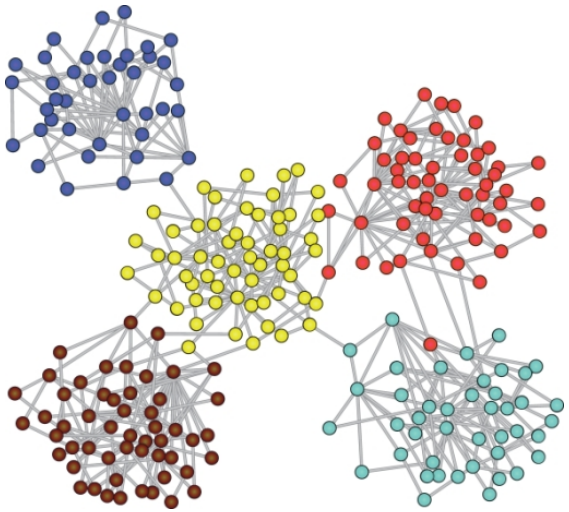
Set S to be all nodes with $\vec{v}_2(i) < 0$, T to be all with $\vec{v}_2(i) \geq 0$.





The Shi-Malik normalized cuts algorithm is one of the most commonly used variants of this approach, using the normalized Laplacian $\overline{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

Important Consideration: What to do when we want to split the graph into more than two parts?



Spectral Clustering:

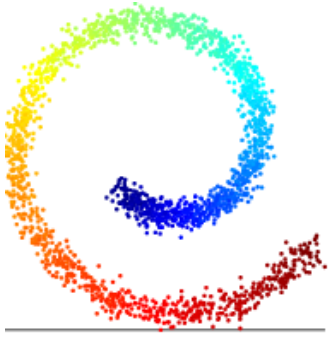
- Compute smallest k nonzero eigenvectors $\vec{v}_{n-1}, \ldots, \vec{v}_{n-k}$ of $\overline{\mathbf{L}}$.

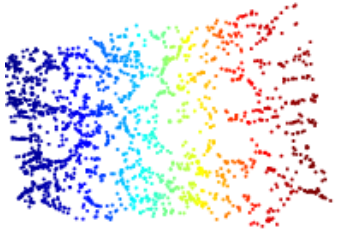
LAPLACIAN EMBEDDING

The smallest eigenvectors of $\mathbf{L} = \mathbf{D} - \mathbf{A}$ give the orthogonal 'functions' that are smoothest over the graph. I.e., minimize

$$\vec{v}^T L \vec{v} = \sum_{(i,j) \in E} [\vec{v}(i) - \vec{v}(j)]^2.$$

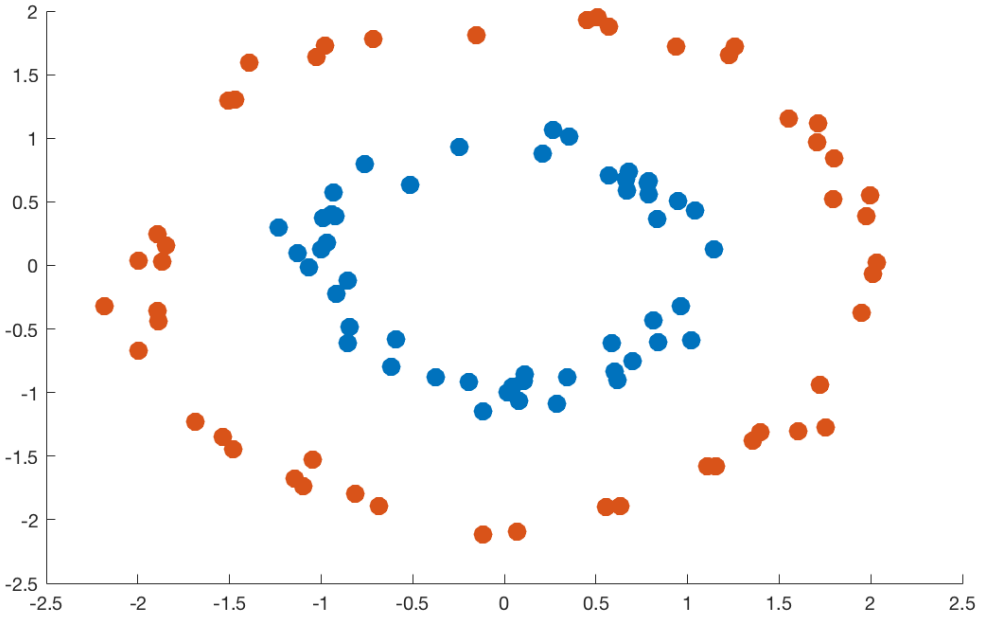
Embedding points with coordinates given by $[\vec{v}_{n-1}(j), \vec{v}_{n-2}(j), \ldots, \vec{v}_{n-k}(j)]$ ensures that coordinates connected by edges have minimum total squared Euclidean distance.





- Spectral Clustering
- Laplacian Eigenmaps
- Locally linear embedding
- Isomap
- Node2Vec, DeepWalk, etc. (variants on Laplacian)

Original Data: (not linearly separable)



k-Nearest Neighbors Graph: